May 2015

Expanding the Knowledge Base of an Interactive 3D Avatar

Author: Benjamin Weinstein
Advised by: S. James Lee
Connecticut College Department of Computer Science

Introduction:

The goal of this project is to examine methods of computer conversation, and to expand the conversational capabilities of a virtual character from a simple mapping system into a realistic two-way conversational ability. The product of this project is a 3-dimensional, virtual avatar that is able to hear a user asking a question, interpret the speech and respond with an acceptable answer. The system will be equipped with decision-making capabilities using Artificial Intelligence concepts.

The character can convert strings to human-like speech. Its speech is accompanied with gestural capabilities. This aids the realistic nature and interactivity of the project.

Semantic analysis and language processing are extremely complex problems. The approach for this project is to use the Artificial Intelligence Markup Language to store the system's knowledge base, using pattern and template matching along with statistical analyses to select the best answer in its correct form. Plenty of work has been done to produce effective implementations of general computer conversation, often done by a team with a depth of time and resources. The goal of this project is to create a system that can learn from non-dialogue texts and is able to converse about the topic of the text.

Methods:

Materials- An Overview of the System

This project was built off of LifeLike, an informative 3D virtual avatar framework on which avatars of various appearances are able to give presentations with a realistic voice and delivery.

The LifeLike system that displays the character was developed in C++, using Microsoft Visual Studio. The avatar is equipped with realistic facial features and it appears on a standard monitor. The avatar has a human-like voice for speaking the output text, as well as gestural capabilities that correspond to the mood carried in the desired speech.

For speech handling, user input is given through a standard microphone (compatible with webcams, standalone microphones, etc.), and interpreted as a string using Google's speech-to-text system, accessed through a Python 2.7 library. The avatar's core script was developed in Python 2.7, to handle dialogue management, and to trigger the avatar to speak.

The knowledge bank of the system is stored in .AIML files, which use an XML format, and are interpreted by the Python 2.7 library PyAIML. Data can be written into the system by hand, can be pulled from the AIML Standard database and can be automatically organized into AIML format and loaded into the system.

Methodology- An Overview of System Implementation

a) Extending the dialog management capabilities of the LifeLike system

Expanding the user experience and the dialogue management system began with switching to Google's speech-to-text system in order to have more robust, reliable and accurate handling of user input for interpretation. The delay in receiving the string from the server was greater than 3 seconds, which began to interfere with the realistic nature of the system, so the core script was changed to a server-client architecture. This was able to increase the efficiency of the system; the server now handles requests for text from audio, and the client triggers avatar actions and received data from the avatar.

b) Examining the intersection of Text-Data Mining and Pattern-Template Matching

Changing the avatar's knowledge base from a simple one-way mapping implementation to a responsive, two-way conversational system through the Artificial Intelligence Markup Language was also an essential step in developing this project further. The avatar's human-like features enhance the Human Computer-Interaction experience, and so does the ability to talk freely with the avatar. With a flexible dialogue system, the user does not have to worry about adhering to predefined content.

In order to implement the intersection of Pattern-Template Matching and Text data mining, an additional Python script was developed to parse text files and assign patterns and templates based on the text content. The system aims to be reliable for topic-based conversation, in which the user and computer communicate about a specific topic (basketball, construction, etc.). To use the information pulled from the text-data effectively, a new AIML file is written for each topic, so that it is possible to supply some preliminary background info to keep the user experience from being frustrating at first. This also allows information from a piece of text to inform the avatar for future conversations, as the previous data can be stored in the system.

Results:

The first goal of this project was to expand the conversational capabilities of the avatar. With an extensive bank of AIML data, the system performs very well, and can be both well informed and informative. The speech-to-text system is reliable, but an extensive

AIML data bank can respond critically to nonsensical answers in the event of a misinterpretation.

The response time of the system is currently fast (under 1s) from the point where is stops listening and retrieves the proper response. However the system gets held up waiting for the user to stop speaking, as the speech-to-text library is written that way. This not only holds up the system, but can also provide unwanted interference if the character is in a public space with background noise. Implementation of a button that the user holds down when speaking can improve the system's overall response time, and diminish the chances of background noise interfering with the system and adding sounds to be converted to text.

The intersection of Text Data Mining and Pattern-Template Matching for computer conversation is one that can be successful. The main drawback of Text-Data Mining is that it requires significant human oversight as it attempts to make connections and obtain information from a bank of text.[2] A major drawback of Pattern-Template Matching is that it generally takes a long time to acquire a dataset large enough to be reliable[2], and with topic-based conversation it is hard to judge how large the dataset needs to be to still be reliable whilst providing answers within the scope of the topic.

The system currently requires significant human oversight. By loading in existing AIML files based on the topic, to give a background basis of knowledge, the system is useable and encourages the user to ask questions, whilst the mining of the text is still incomplete. The ease of populating AIML files with information from the text once it is targeted shows promise of the system.

Discussion:

Furthering this work with specific next steps can allow this project to deeply understand the intersection of Text-Data Mining and Pattern-Template Matching.

Implementation of an oversight and correction system will aid knowledge acquisition. The user can select one of three options when the system returns an answer:

1. Mark as correct.
2. Mark as incorrect. The user is then prompted to provide a correct answer. The system replaces the old answer with the new, correct one.
3. Mark as accurate but not ideal. The user is asked to rate the old answer's accuracy on a one to ten scale, and then to provide a better answer. The system keeps the old answer in the system if it has a high accuracy rating, and it stores the new answer in the system.

In addition to improving the accuracy of responses and giving the system a basis to improve over time, this addition can also provide metrics on the system's performance, as the number of corrections based on the length of the text.

A user study can be used to for either gauging the effectiveness of the system in creating an engaging user experience or tracking the effectiveness of the system over time. When the system improves over time, it can have applications in automated learning through, for example, a web scraper.

References:

[1] A. P. Saygin and I. Cicekli, “Pragmatics in human-computer conversations,” *Journal of Pragmatics*, vol. 34, pp. 227-258, December 2002.

[2] R. P Schumaker, Y. Liu, M. Ginsburg and H. Chen, “Evaluating mass knowledge acquisition using the ALICE chatterbot: The AZ-Alice dialog system,” *International Journal of Human-Computer Studies*, vol. 64, no. 11, pp. 1132-1140, November 2006.

[3] J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu and A. Stent, “Toward conversational human-computer interaction,” *AI Magazine*, vol. 22, no. 4, pp. 27-37, December 2001.